combination of the IP address of the station and a port number determines a socket. One computer writes data to a socket in order to send the data to a second computer, and the second computer reads from its socket the data. This can be illustrated as a telephone call. In order to speak with a subscriber, one has to dial its telephone number (in our case the IP address), and then its extension (the port number). After the connection has been established, one talks to its handset (socket), and the other listens through his handset (socket).

[0044] A well-known port refers herein to a protocol port that is widely used for a certain type of data on the network. For example, HTTP is typically assigned port **80**, FTP transfer is port **21**, the POP3 the port number **110**, and X-Windows 6000. A Privileged port refers herein to a protocol port numbered from **1** through **1023**.

[0045] Under a Unix-based operating system, the FTP server is a process that holds a communication session through a well-known port. During the communication session, the FTP server performs commands as defined in RFC 959. RFC (Request For Comment) is a formal document from the Internet Engineering Task Force (IETF) that is the result of committee drafting and subsequent review by interested parties.

[0046] The first two commands usually to be performed are the "USER" and the "PASS" commands, which are used for identifying the user name and its password to the FTP server. The user identification is carried out prior to any further commands to the FTP server, such as downloading or uploading files. The FTP server validates the name and the password using the computer's username and password ("/etc/passwd" in a Unix-based systems), and if they correspond—the user is allowed to access files according to the specific user's permissions on the file system. Of course, any other secure mechanism can be used, and this scheme alone has been described herein for the sake of brevity.

[0047] Inetd (INternET Daemon) is a Unix function that manages many common TCP/IP services. It is activated at startup, waits for various connection requests (FTP, Telnet, etc.) and launches the appropriate server components.

[0048] According to the prior art, an FTP server or any other Internet server can be activated in two modes:

[0049] The "Inetd" mode, in which a single process (the Inetd daemon) serves a plurality of network services. The daemon "listens" on specific ports, waiting for requests for connection. When a request for connection arrives, it creates the service process (according to the well-known port), and allows it to handle the communication session.

[0050] The benefit of this approach is that the system resources are saved since there is a single process that listens on all the ports instead of a plurality of processes.

[0051] The "Stand-alone" mode, in which the relevant process is created once (for example, when the system boots), and the process handles the connections. This mode suits services that typically are active all the time and therefore starting and terminating a process every time a connection arrives results in an unnecessary overhead.

[0052] Usually, HTTP server operates in the Stand-alone mode, while the FTP server, POP3 server, and SMTP server operate in the "Inetd" mode.

The Problem of Name-Based FTP and E-mail

[0053] In the original design of the FTP as described in RFC 959, the FTP was directed to serving one domain, associated with one IP address, unlike the HTTP service in which the commands contain the object domain. Hence, if, for example, two domains ftp.aaa.com and ftp.bbb.com reside on the same hosting computer and share the same IP address, then the users xxx@ftp.aaa.com and xxx@ftp.bbb.com are the same. Moreover, if a user tries to access "ftp.aaa.com/pub/", he will reach the same directory as "ftp.bbb.com/pub/".

[0054] Those skilled in the art might determine a partial solution to this issue, by defining specific sub-directories forftp.aaa.com and for ftp.bbb.com, and prevent the users of each domain accessing the other directory. However, there are several problems inherent in such an approach:

[0055] The users would know that there is another directory, but they would not be able to access it. This might be a drawback in Web hosting companies, as it is preferable that each domain be unaware of the existence of other domains residing on the same hosting computer.

[0056] Only one user with a specific name ("xxx" in the above example) can exist on a hosting computer, and therefore common names (like "webmaster", "jobs", "info" etc.) cannot be allocated to a plurality of domains. It should be noted that in order to support this solution, a system should keep the users of each domain in a separate location. For example, the VDS technology, as described in copending Israeli Patent Application No. 147560, is an example of such a solution.

[0057] A FTP server is used for all the domains hosted by a hosting computer. Therefore, there is no trivial way to calculate the resources consumed by a domain, as the process resources are shared.

[0058] By changing the permissions of a file, the file may become accessible to other users, such as "cookie" users.

The Solution

[0059] Actually, the basis of the problem described hereinabove is the protocol, which is not adapted for providing FTP services from a singular computer to a plurality of domains. The same problem relates to mail services as well. Due to the limitation in the protocol, providers of services are limited as well.

[0060] The problem of the "missing" domain applies only to POP3. In SMTP the domain is passed as well, and therefore there is no need to add the domain as part of the user name, as for FTP. However, the wrapper is still required for handling the requests by the correct server, i.e., the server of the appropriate VDS.

[0061] According to the solution disclosed herein, an intermediary between the client and the servers is added in the communication chain. The role of the intermediary is to identify in a request for service the domain of the request, and to direct the request to this server in a standard form.

[0062] Thus, the intermediary interacts with the client as it were the service provider, and with the service provider as it were the client.